

 Mon parcours PROJET EN COURS

# Résolvez des problèmes en utilisant des algorithmes en Python

20%

**Mission** Cours Ressources Évaluation 60 heures

Mis à jour le jeudi 5 mai 2022

Le nommage des livrables à déposer sur la plateforme a été changé et des indications sur les temps de soutenance ont été ajoutées - 20/04/2022.

## Scénario

Vous venez de rejoindre **AlgoInvest&Trade**, une société financière spécialisée dans l'investissement. La société cherche à optimiser ses stratégies d'investissement à l'aide d'algorithmes, afin de dégager davantage de bénéfices pour ses clients.



Le logo de l'entreprise

Vous avez passé vos premiers jours à rencontrer votre petite équipe de six personnes. Votre responsable technique, Robin, a expliqué que même si tous les membres de l'équipe connaissent les termes techniques, ils sont en revanche tous issus de l'économie et de la finance, plutôt que de l'informatique. Comme vous êtes un des seuls développeurs, votre rôle sera principalement de traduire leurs besoins commerciaux en solutions techniques.

Un matin, vous arrivez au travail et vous recevez le courriel suivant :

## Sujet : Algorithme pour maximiser nos bénéfices

Bonjour,

J'espère que vous êtes prêt pour votre premier vrai projet ! Nous avons besoin d'aide pour optimiser les solutions de nos clients afin de rendre nos programmes d'investissement à court terme plus compétitifs. Nous avons besoin que vous conceviez un algorithme qui maximisera le profit réalisé par nos clients après deux ans d'investissement. Votre algorithme doit suggérer une liste des actions les plus rentables que nous devrions acheter pour maximiser le profit d'un client au bout de deux ans.

Nous avons les contraintes suivantes :

- Chaque action ne peut être achetée qu'une seule fois.
- Nous ne pouvons pas acheter une fraction d'action.
- Nous pouvons dépenser au maximum 500 euros par client.

Vous trouverez ci-dessous une liste des actions sur lesquelles nous travaillons :

Actions #	Coût par action (en euros)	Bénéfice (après 2 ans)
Action-1	20	5%
Action-2	30	10%
Action-3	50	15%
Action-4	70	20%
Action-5	60	17%
Action-6	80	25%
Action-7	22	7%
Action-8	26	11%
Action-9	48	13%
Action-10	34	27%
Action-11	42	17%
Action-12	110	9%
Action-13	38	23%

Action-14	14	1%
Action-15	18	3%
Action-16	08	8%
Action-17	04	12%
Action-18	10	14%
Action-19	24	21%
Action-20	114	18%

Je sais que vous êtes nouveau dans le monde de la finance, alors voici un aperçu de la signification de chaque colonne :

- **Actions** – Chaque "Action-#" représente une action dans une entreprise différente. Si vous imaginez la valeur d'une entreprise comme étant une tarte entière, chaque action est comme une part de cette tarte.
- **Coût par action (en euros)** – Le coût d'une action de l'entreprise en euros.
- **Bénéfice (après 2 ans)** – Il s'agit du bénéfice réalisé par le titulaire de l'action après 2 ans d'investissement dans l'entreprise. Le bénéfice est un pourcentage du coût de l'action.

Parce que nous voulons être aussi transparents que possible pour nos clients, nous voulons que le programme essaie toutes les différentes combinaisons d'actions qui correspondent à nos contraintes, et choisisse le meilleur résultat. Le programme doit donc lire un fichier contenant des informations sur les actions, explorer toutes les combinaisons possibles et afficher le meilleur investissement.

Merci !

Robin Greene

Tech Lead, AlgoInvest&Trade

Après avoir lu les exigences, vous commencez à développer une solution de force brute et vous envoyez le code Python à Robin dans un fichier ("bruteforce.py").

## Partie 2: Optimisation d'algorithme

Quelques jours plus tard, vous recevez le courriel suivant :

Bonjour,

Vous vous êtes bien débrouillé, bravo. La solution que vous avez fournie pour l'ensemble de données initial fonctionne très bien, et nous sommes tous ravis de commencer à utiliser davantage d'algorithmes dans notre processus de décision. Nous aimerions nous engager sur de nouveaux marchés financiers, et nous avons maintenant un nombre beaucoup plus important d'actions dans notre ensemble de données.

Il semble que le programme que vous avez créé prenne beaucoup de temps et de ressources pour choisir la meilleure opportunité d'investissement. Nos clients sont des personnes très occupées, et préfèrent une production beaucoup plus rapide. Le programme doit fournir une réponse en moins d'une seconde.

Nous avons discuté avec nos clients et ils font confiance à AlgoInvest&Trade pour prendre les bonnes décisions. Nous n'avons plus besoin d'explorer toutes les combinaisons possibles, ce qui, à mon avis, devrait accélérer l'algorithme.

Sienna, l'un de nos principaux conseillers financiers, est très intéressée par votre programme. Elle a un peu d'expérience en matière de développement et aimerait en savoir plus sur votre travail. Pourriez-vous préparer une présentation de diapositives, et me l'envoyer avec votre programme optimisé ?

Voici la liste des livrables qu'elle a demandés :

- Le programme Python optimisé, qui lit un fichier contenant des informations sur les actions, et fournit la meilleure stratégie d'investissement.
- Un jeu de diapositives contenant les éléments suivants :
  - une analyse de votre algorithme de force brute ;
  - un diagramme, un organigramme ou un pseudocode décrivant le processus de réflexion qui sous-tend la solution optimisée ;
  - l'algorithme choisi pour la version optimisée, et les limites de l'algorithme (cas limites) ;
  - une comparaison de l'efficacité et des performances de l'algorithme de force brute par rapport à l'algorithme optimisé en utilisant la notation Big-O, la complexité temporelle et l'analyse de la mémoire.

J'espère que cette liste a un sens pour vous !

Merci encore,

Robin Greene

Tech Lead, AlgoInvest&Trade

Il y a beaucoup à faire ! Vous commencez à créer un nouveau fichier, "optimized.py", pour votre nouveau programme.

## Partie 3 : backtesting et optimisation

Bonjour,

Sienna a été très impressionnée par vos diapositives. Elle aimerait utiliser votre programme avec davantage de clients. Cependant, elle s'inquiète de l'exactitude du programme. Elle aimerait tester sa précision en l'exécutant sur des ensembles de données antérieurs, et en vérifiant les résultats de votre algorithme.

Les données sont disponibles [ici \(1\)](#) et [ici \(2\)](#), et les décisions d'investissement correspondantes de Sienna sont également disponibles [ici \(1\)](#) et [ici \(2\)](#). Elle m'a averti qu'il s'agit de fichiers historiques qu'elle a elle-même construits il y a quelques années, et qu'il peut y avoir des données manquantes ou incorrectes !

Pourriez-vous préparer un jeu de diapositives contenant une comparaison côte à côte entre les résultats de votre algorithme et les choix de Sienna ? Elle m'a dit qu'il faudrait créer un rapport d'exploration de l'ensemble des données. Elle vous rencontrera pour revoir la présentation.

Merci encore,

Robin Greene

Tech Lead, AlgoInvest&Trade

Vous téléchargez les ensembles de données et commencez immédiatement à préparer la réunion. En discutant avec vos collègues, vous avez appris qu'elle possède une solide expérience en matière de science des données et d'algorithmes, et qu'elle a même écrit quelques programmes Python il y a quelques années. Vous avez intérêt à être prêt !

## Livrables

1. Un **fichier Python** `bruteforce.py` avec la solution de force brute.
2. Un **fichier Python** `optimized.py` avec la version optimisée de l'algorithme.
3. Un **jeu de diapositives** au format PDF expliquant ce qui suit (20 diapositives maximum) :
  - la solution optimisée, y compris :
    - l'analyse de l'algorithme de force brute,
    - un diagramme/agenda/pseudocode,
    - l'algorithme choisi et ses limites,
    - l'analyse des performances et de l'efficacité de vos algorithmes ;
  - une comparaison côte à côte entre la sortie de votre algorithme et les choix de Sienna, via un rapport d'exploration de l'ensemble des données.

Pour faciliter votre passage devant le jury, déposez sur la plateforme, dans un dossier zip nommé "**Titre\_du\_projet\_nom\_prénom**", tous les livrables du projet comme suit : **Nom\_Prénom\_n° du livrable\_nom du livrable\_\_date de démarrage du projet**. Cela donnera :

- `Nom_Prénom_1_bruteforce_mmaaaa ;`

- *Nom\_Prénom\_2\_optimized\_mmaaaa ;*
- *Nom\_Prénom\_3\_diapositives\_mmaaaa.*

Par exemple, le premier livrable peut être nommé comme suit :

*Dupont\_Jean\_1\_bruteforce\_012022*

## Soutenance

Durant la présentation orale, l'évaluateur interprétera le rôle de Sienna. La soutenance est structurée de la manière suivante :

- **Présentation des livrables (15 minutes)**
  - Présentation de vos algorithmes **(10 minutes)** :
    - Comparez votre solution bruteforce avec la solution optimisée.
    - Soyez prêt à répondre aux questions suivantes :
      - Pourquoi faut-il autant de temps pour fournir une réponse avec un ensemble de données plus important ?
      - Votre algorithme n'a pas à calculer toutes les réponses possibles. Comment y êtes-vous parvenu ?
  - Présentation des résultats de votre algorithme **(5 minutes)** :
    - Rapport d'exploration des ensembles de données.
    - Présentation des résultats de votre algorithme, et comparaison critique avec celui de Sienna.
- **Discussion (10 minutes)**
  - L'évaluateur jouera le rôle de Sienna. Il vous challengera sur votre méthode et vos livrables.
- **Debrief (5 minutes)**
  - À la fin de la soutenance, l'évaluateur arrêtera de jouer le rôle de Sienna, pour vous permettre de débriefer ensemble.

Votre présentation devrait durer 15 minutes (+/- 5 minutes). Puisque le respect des durées des présentations est important en milieu professionnel, les présentations en dessous de 10 minutes ou au-dessus de 20 minutes peuvent être refusées.

## Compétences évaluées



Déconstruire un problème



Développer un algorithme pour résoudre un problème

[OPENCLASSROOMS](#)[OPPORTUNITÉS](#)[AIDE](#)[POUR LES ENTREPRISES](#)[EN PLUS](#)[Français](#)

Télécharger dans  
**l'App Store**

